Portofolio 3

Gustav Helms, Sigrid Agersnap, Anders Hjulmand, Morten Street

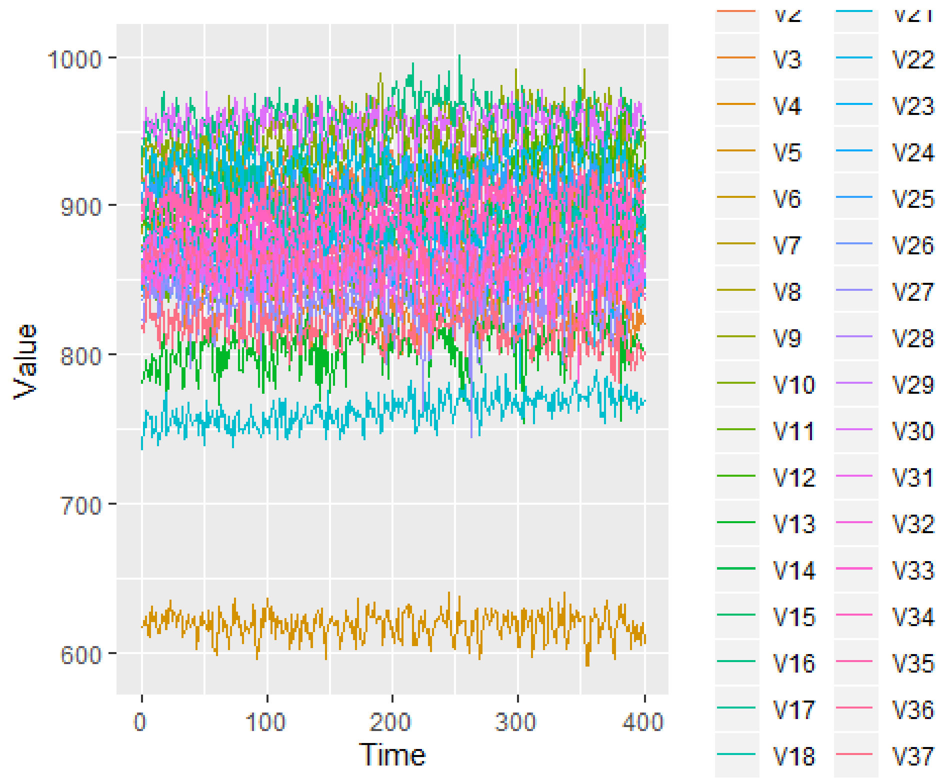
2/19/2020

```
#Setup
library(pacman)
pacman::p_load("tidyverse", "lme4", "reshape2", "pracma")

#load data

fmri<-as.matrix(read.csv("portfolio_assignment3_aud_fmri_data37.csv",
header=FALSE))
##making it a time-series
fmri2<-ts(fmri)
##design


fmrides<-as.matrix(read.csv("portfolio_assignment3_aud_fmri_design.csv",
header=FALSE))
##making it a time-series
fmrides2<-ts(fmrides)
```

1. Make three figures:

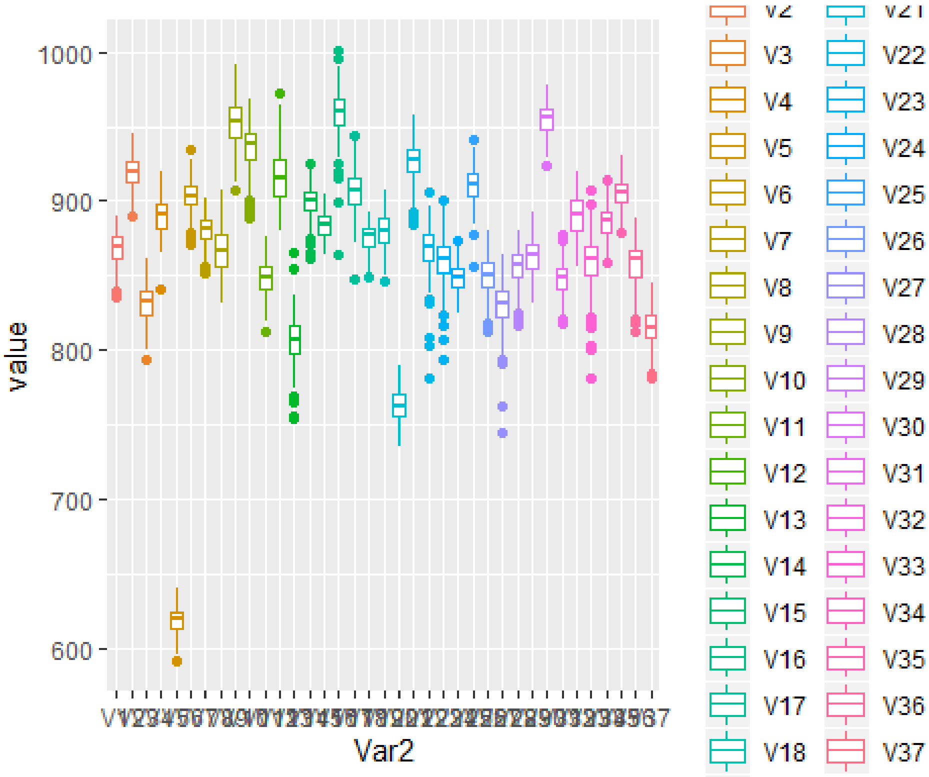
1.a. A figure with lineplots of the data from all participants as a function of time in one figure.

```
# Restructuring the df
melt <- melt(fmri2)

ggplot(melt, aes(melt$Var1, melt$value, colour = melt$Var2))+
  geom_line()+
  labs(x = "Time", y = "Value")
```



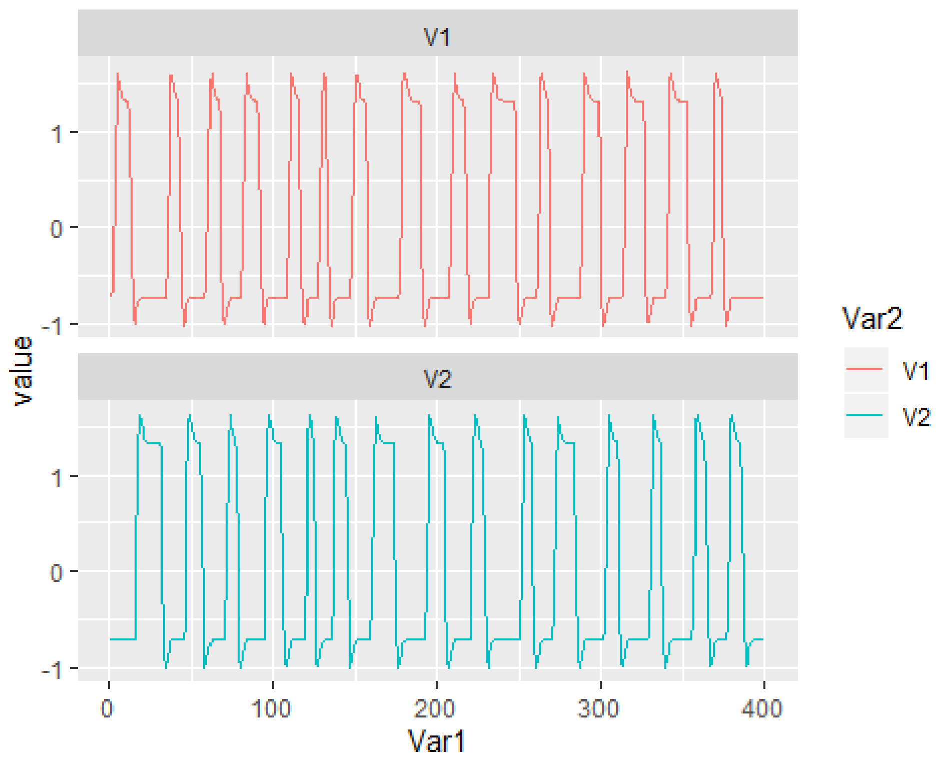
1.b. A boxplot with the signal intensity for each participant. Note how much the baseline signal can vary between participants.

```
ggplot(melt, aes(Var2, value, colour = Var2))+
  geom_boxplot()
```



1.c. A lineplots figure with the model covariates.

```
melt2 <- melt(fmrides2)

ggplot(melt2, aes(Var1, value, colour = Var2))+
  geom_line()+
  facet_wrap(melt2$Var2, ncol = 1)
```



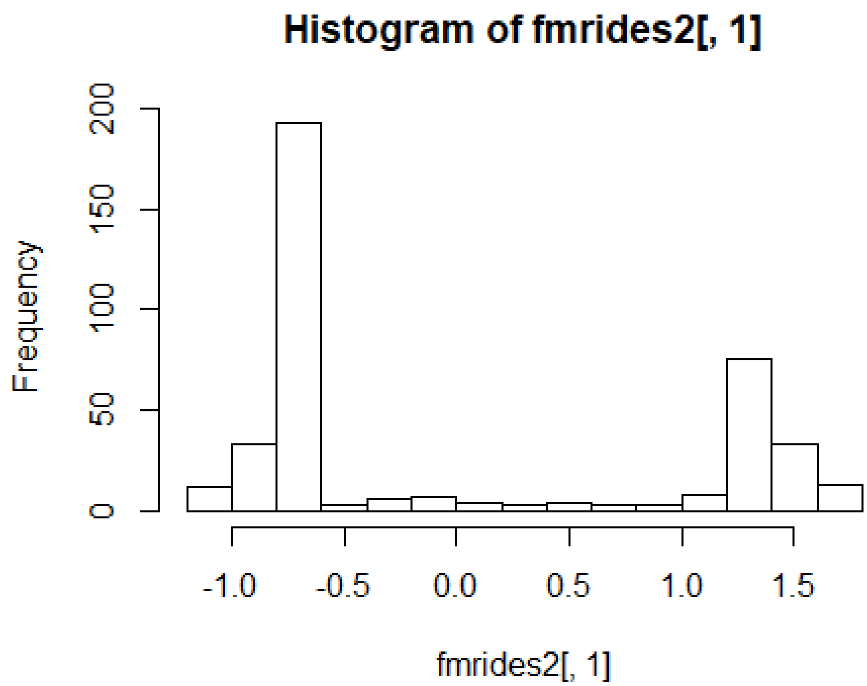
2. Based on the shape of the model: How many stories did the participants listen to in each condition (you can also automatise this, e.g. using "findpeaks" in library(pracma))?

```
#Finding peaks in condition 1
nrow(findpeaks(fmrides[,1]))

## [1] 15

# There are 15 peaks

#Finding peaks in condition 2
nrow(findpeaks(fmrides[,2]))

## [1] 15

# There are 15 peaks
```

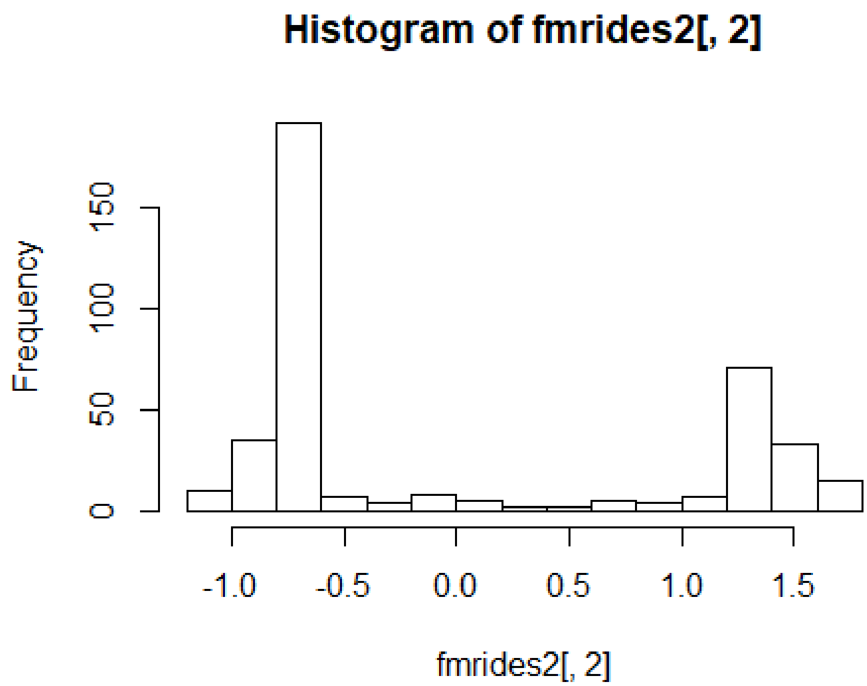
There was 15 stories in each condition. In total that's 15+15=30 stories.

3.a. Are the two model covariates correlated?

```
#Checking visually for normality in the data
hist(fmrides2[,1])
```




```
#By visual inspection the data are not normally distributed

#Checking visually for normality in the data
hist(fmrides2[,2])
```




```
#By visual inspection the data are not normally distributed
```

Since the data are not normally distributed a non-parametric correlation test is used.

```
# Running correlationtest
cor <- cor.test(fmrides2[,1],fmrides[,2], method = "spearman")

## Warning in cor.test.default(fmrides2[, 1], fmrides[, 2], method =
## "spearman"): Cannot compute exact p-value with ties

cor

##
##  Spearman's rank correlation rho
##
## data:  fmrides2[, 1] and fmrides[, 2]
## S = 16863964, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.5810065
```

There is a significant negative correlation of rho=-0.58, p< 0.001, between the two conditions which reflects a middle effect. This correlation might reflect that the two conditions have a similar trend which is skewed from each other (see figure in 1.c).

3.b. Have the covariates been mean-centered?

```
# Sum of covariate 1
sum(fmrides2[,1])

## [1] -0.0005395

# Sum of covariate 2
sum(fmrides2[,2])

## [1] 0.000818
```

Both sums of the covariates approximates 0, meaning they are mean-centered.

4. Please report the percentage of shared variance in the two covariates.

```
# The r^2 of the correlation test exlains how much of the variance the
covariates explains.
cor$estimate^2

##       rho
## 0.3375686
```

The two covariates share 33.8 % of the variance in the data.

5. Pick one participant's data set.

```
# Subsetting participant 23.
p1 <- fmri2[,23]
```

Conduct 6 analyses using lm(): ### 5.a. Fit the model as it is, including intercept.

```
# Doing lm
summary(lm(p1 ~ fmrides2))

##
## Call:
## lm(formula = p1 ~ fmrides2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.406  -9.152   0.349   8.344  37.720
##
## Coefficients:
##             Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 860.5850     0.6878 1251.304  < 2e-16 ***
## fmrides2V1    5.8274     0.8199    7.107 5.53e-12 ***
## fmrides2V2    5.1652     0.8199    6.300 7.92e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.76 on 397 degrees of freedom
```

```
## Multiple R-squared:  0.1293, Adjusted R-squared:  0.1249
## F-statistic: 29.48 on 2 and 397 DF,  p-value: 1.152e-12
```

Both covariates significantly explains the data with p-values<0.001.

5.b. Fit the model as it is, excluding intercept.

```r
#Mean centering the data to remove the intercept
p1_mean <- p1-mean(p1)

# making the new model, excluding the intercept
summary(lm(p1_mean ~ fmrides2))

##
## Call:
## lm(formula = p1_mean ~ fmrides2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.406  -9.152   0.349   8.344  37.720
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.703e-06  6.878e-01   0.000        1
## fmrides2V1   5.827e+00  8.199e-01   7.107 5.53e-12 ***
## fmrides2V2   5.165e+00  8.199e-01   6.300 7.92e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.76 on 397 degrees of freedom
## Multiple R-squared:  0.1293, Adjusted R-squared:  0.1249
## F-statistic: 29.48 on 2 and 397 DF,  p-value: 1.152e-12
```

The slopes stay the same but the intercept approximates 0 in the new model with mean centered data.

5.c. Fit only the 1st covariate as a model.

```r
#Fitting only the first covariate
m1 <- summary(lm(p1 ~ fmrides2[,1]))
m1

##
## Call:
## lm(formula = p1 ~ fmrides2[, 1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -64.389  -8.383  -0.127   9.599  42.611
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
```

```
## (Intercept)   860.5850     0.7204 1194.592  < 2e-16 ***
## fmrides2[, 1]   3.0237     0.7213    4.192 3.41e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.41 on 398 degrees of freedom
## Multiple R-squared:  0.04228,    Adjusted R-squared:  0.03988
## F-statistic: 17.57 on 1 and 398 DF,  p-value: 3.411e-05
```

Using the 1st covariate as a model significantly explains the data.

5.d. Fit only the 2nd covariate as a model.

```
#Fitting only the second covariate
m2 <- summary(lm(p1 ~ fmrides2[,2]))
m2

##
## Call:
## lm(formula = p1 ~ fmrides2[, 2])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.319  -8.598   0.779   8.854  41.854
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)   860.5850     0.7293 1180.052  < 2e-16 ***
## fmrides2[, 2]   2.0021     0.7302    2.742  0.00639 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.59 on 398 degrees of freedom
## Multiple R-squared:  0.01854,    Adjusted R-squared:  0.01607
## F-statistic: 7.518 on 1 and 398 DF,  p-value: 0.006385
```

Using the 2nd covariate as a model significantly explains the data. However, the slope gradient are different in the two models.

The residuals represent the variance left when fitting a model. They are thus data that have been “cleaned” from the variance explained by the model. We can use those “cleaned” data to fit another model on. This is similar to using a type III sum of squares approach to your statistics.

5.e. Fit the 2nd covariate to the residuals from analysis 5.c., the 1st covariate only analysis

```
#Fitting the 2nd covariate to the residuals of the first covariate
summary(lm(m1$residuals~fmrides2[,2]))

##
## Call:
```

```
## lm(formula = m1$residuals ~ fmrides2[, 2])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -69.365  -9.141   0.929   8.907  38.854
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.451e-06  6.969e-01   0.000        1
## fmrides2[, 2]  3.643e+00  6.978e-01   5.221 2.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.94 on 398 degrees of freedom
## Multiple R-squared:  0.0641, Adjusted R-squared:  0.06175
## F-statistic: 27.26 on 1 and 398 DF,  p-value: 2.871e-07
```

5.f. Fit the 1st covariate to the resistuals from 5.d., the 2nd covariate only analysis

```
#Fitting the 1st covariate to the residuals of the second covariate
summary(lm(m2$residuals~fmrides2[,1]))
```

```
##
## Call:
## lm(formula = m2$residuals ~ fmrides2[, 1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.333  -8.397   0.292   9.304  40.716
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.544e-06  6.996e-01   0.000        1
## fmrides2[, 1] 4.110e+00  7.005e-01   5.868 9.31e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.99 on 398 degrees of freedom
## Multiple R-squared:  0.07962,    Adjusted R-squared:  0.07731
## F-statistic: 34.43 on 1 and 398 DF,  p-value: 9.312e-09
```

5.g. Does the order in which the predictor variables are fitted to the data matter for the estimates? If it does, what can explain this?

The order in which the predicter variables are fitted to the data matters for the estimate. Fitting the residuals from the first model to the second covariate had an estimate of beta=3.643, and vice versa the other model had an estimate of beta=4.11. The difference of 0.5 in the gradients of the two covariates could be explained by the skewness of the two conditions in the design matrix (see figure in 1.c).

6. Fit the full model to each of the 37 participants' data and extract the coefficients for each participant. (hint: the full participant data frame can be set as outcome. Alternatively, you can change the data structure and use lmList from assignement 1 (remember pool=FALSE)).

```
#Making the full model
m <- lm(fmri2~fmrides2)

#Organizing the data into a data frame
df <- data.frame("Intercept" = m$coefficients[1,],
                 "V1" = m$coefficients[2,],
                 "V2" = m$coefficients[3,])
df
```

```
##     Intercept       V1       V2
## V1   867.3275 9.582330 8.926516
## V2   919.4350 5.773264 6.104578
## V3   831.2300 7.340939 8.233871
## V4   890.1325 2.682228 3.133700
## V5   618.7725 5.872990 6.041478
## V6   903.3950 3.209499 3.230518
## V7   880.8425 2.606846 2.499852
## V8   868.2550 3.828594 3.530697
## V9   952.3775 2.422324 0.992906
## V10  936.6850 6.400593 7.207420
## V11  847.8800 6.989003 6.538979
## V12  916.2025 3.464375 3.385156
## V13  806.2725 4.626842 3.595325
## V14  900.0700 3.304558 3.914555
## V15  884.1000 3.566603 3.495208
## V16  959.1700 5.566010 5.802205
## V17  907.0200 3.756718 3.873774
## V18  875.3125 5.436519 5.276158
## V19  879.4675 2.814987 2.626577
## V20  762.8750 2.829585 2.658036
## V21  926.4350 7.351865 6.394320
## V22  868.1425 5.963731 5.905760
## V23  860.5850 5.827429 5.165241
## V24  848.9875 3.880726 4.050404
## V25  910.4025 6.534162 6.399795
## V26  850.3925 4.849484 4.703407
## V27  829.9300 4.997005 3.951056
## V28  855.8500 7.818241 7.767283
## V29  863.1775 5.474936 5.903824
## V30  954.6175 6.648170 6.628171
## V31  847.9225 5.056695 5.756162
## V32  889.6675 4.891110 4.607988
```

```
## V33  859.4825  4.526764  6.001619
## V34  885.8825  4.904306  3.930084
## V35  905.4525  4.095500  4.176857
## V36  857.7000 10.521195 10.631068
## V37  815.1875  4.760363  4.646695
```

6.a. Test the two individual hypotheses that the set of coefficient from each covariate is different from zero across the whole group (similar to assignment 1).

```
# doing one sample t-test for covariate 1.
t.test(df$V1)

## 
##  One Sample t-test
## 
## data:  df$V1
## t = 16.607, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.512224 5.767586
## sample estimates:
## mean of x 
##  5.139905
```

Covariate 1 is significantly different from 0, t(36)=16.61, p<0.001.

```
# doing one sample t-test for covariate 2.
t.test(df$V2)

## 
##  One Sample t-test
## 
## data:  df$V2
## t = 15.603, df = 36, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  4.413274 5.731982
## sample estimates:
## mean of x 
##  5.072628
```

Covariate 2 is significantly different from 0, t(36)=15.6, p<0.001.

6.aa. Make a contrast that investigates the difference between the two covariates, i.e. the two types of stories (hint: subtraction).

```
df$Contrast <- df$V1-df$V2
df

##     Intercept        V1        V2    Contrast
## V1   867.3275  9.582330  8.926516  0.65581370
```

```
## V2   919.4350  5.773264  6.104578 -0.33131441
## V3   831.2300  7.340939  8.233871 -0.89293270
## V4   890.1325  2.682228  3.133700 -0.45147183
## V5   618.7725  5.872990  6.041478 -0.16848855
## V6   903.3950  3.209499  3.230518 -0.02101973
## V7   880.8425  2.606846  2.499852  0.10699381
## V8   868.2550  3.828594  3.530697  0.29789701
## V9   952.3775  2.422324  0.992906  1.42941758
## V10  936.6850  6.400593  7.207420 -0.80682690
## V11  847.8800  6.989003  6.538979  0.45002325
## V12  916.2025  3.464375  3.385156  0.07921988
## V13  806.2725  4.626842  3.595325  1.03151654
## V14  900.0700  3.304558  3.914555 -0.60999652
## V15  884.1000  3.566603  3.495208  0.07139554
## V16  959.1700  5.566010  5.802205 -0.23619571
## V17  907.0200  3.756718  3.873774 -0.11705603
## V18  875.3125  5.436519  5.276158  0.16036159
## V19  879.4675  2.814987  2.626577  0.18841019
## V20  762.8750  2.829585  2.658036  0.17154882
## V21  926.4350  7.351865  6.394320  0.95754501
## V22  868.1425  5.963731  5.905760  0.05797112
## V23  860.5850  5.827429  5.165241  0.66218773
## V24  848.9875  3.880726  4.050404 -0.16967800
## V25  910.4025  6.534162  6.399795  0.13436743
## V26  850.3925  4.849484  4.703407  0.14607779
## V27  829.9300  4.997005  3.951056  1.04594858
## V28  855.8500  7.818241  7.767283  0.05095722
## V29  863.1775  5.474936  5.903824 -0.42888830
## V30  954.6175  6.648170  6.628171  0.01999890
## V31  847.9225  5.056695  5.756162 -0.69946688
## V32  889.6675  4.891110  4.607988  0.28312193
## V33  859.4825  4.526764  6.001619 -1.47485532
## V34  885.8825  4.904306  3.930084  0.97422218
## V35  905.4525  4.095500  4.176857 -0.08135714
## V36  857.7000 10.521195 10.631068 -0.10987261
## V37  815.1875  4.760363  4.646695  0.11366801
```

6.b. Test the hypothesis that the contrast is different from zero across participants.

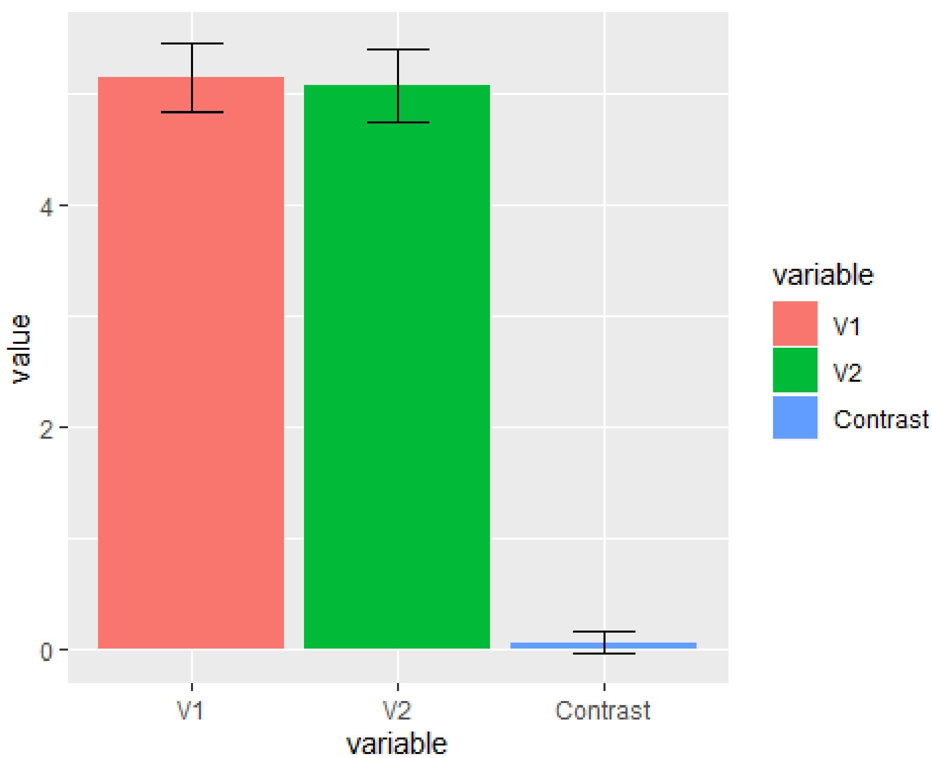
```
t.test(df$Contrast)

## 
##  One Sample t-test
## 
## data:  df$Contrast
## t = 0.69615, df = 36, p-value = 0.4908
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1287197  0.2632734
## sample estimates:
```

```
##  mean of x
## 0.06727684
```

The contrast between covariate 1 and 2 is not significantly different from 0, t(36)=0.25, p=0.80. This entails that the slopes of the covarients across participants are not significantly different.

6.c. Make a bar diagram including the mean effect of the two coefficents and the contrast, including error bars (indicating standard error of mean).

```
#Rearranging the df
melt_df <- melt(df)

## No id variables; using all as measure variables

#making the bar-plot, excluding intercept
ggplot(melt_df %>% filter(variable != "Intercept"), aes(variable,value, fill
= variable))+
  stat_summary(fun.y = mean, geom = "Bar")+
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.3)
```



We do not see a significant difference in the mean effect of the two covariants, i.e. there is no difference between the two conditions (factual or fiction). This is seen by the two error-bars overlapping, and further underpinned by the t-test in 6.b.

7.a. For each partipant, add a covariate that models the effect of time (hint: 1:400).

```
#Making a new column in our design matrix that models the effect of time from
1 to 400.
fmrides2 <- cbind(fmrides2, "Time" =c(1:400))
```

7.b. Does that improve the group results in term of higher t-values?

```
#Making the whole analysis aging

#Making the full model
m <- lm(fmri2~fmrides2)

#Organizing the data into a data frame
df1 <- data.frame("Intercept" = m$coefficients[1,],
                  "V1" = m$coefficients[2,],
                  "V2" = m$coefficients[3,],
                  "Time" = m$coefficients[4,]
               )
df1

##     Intercept       V1       V2          Time
## V1   868.0578 9.550016 8.884649 -0.0036422065
## V2   917.2425 5.870281 6.230279  0.0109351864
## V3   837.5525 7.061170 7.871389 -0.0315337457
## V4   876.7980 3.272271 3.898191  0.0665059681
## V5   618.5361 5.883448 6.055029  0.0011788024
## V6   900.4885 3.338111 3.397156  0.0144963991
## V7   880.7089 2.612759 2.507513  0.0006664460
## V8   848.4977 4.702847 4.663425  0.0985402495
## V9   941.9349 2.884405 1.591603  0.0520828978
## V10  927.6008 6.802563 7.728234  0.0453075849
## V11  847.9025 6.988008 6.537690 -0.0001121534
## V12  892.7402 4.502570 4.730295  0.1170187452
## V13  795.6141 5.098470 4.206392  0.0531589673
## V14  899.2625 3.340289 3.960850  0.0040273769
## V15  880.0330 3.746568 3.728380  0.0202845108
## V16  954.4670 5.774115 6.071837  0.0234562781
## V17  912.1634 3.529123 3.578890 -0.0256530483
## V18  874.1432 5.488261 5.343197  0.0058319930
## V19  868.4097 3.304290 3.260543  0.0551511340
## V20  751.2437 3.344263 3.324880  0.0580112666
## V21  925.7719 7.381207 6.432338  0.0033073449
## V22  871.8300 5.800563 5.694350 -0.0183913104
## V23  860.1728 5.845666 5.188871  0.0020556440
## V24  850.5987 3.809432 3.958032 -0.0080357482
## V25  903.3099 6.848006 6.806427  0.0353744407
## V26  838.8855 5.358663 5.363126  0.0573914391
## V27  835.1618 4.765501 3.651108 -0.0260936683
## V28  859.9679 7.636024 7.531194 -0.0205382764
```

```
## V29  863.3169  5.468770  5.895835 -0.0006950470
## V30  952.5385  6.740166  6.747366  0.0103691595
## V31  846.4594  5.121437  5.840045  0.0072972862
## V32  873.1977  5.619891  5.552235  0.0821435352
## V33  861.8084  4.423846  5.868273 -0.0116002920
## V34  892.3598  4.617687  3.558725 -0.0323059488
## V35  900.5387  4.312931  4.458572  0.0245074791
## V36  864.3029 10.229020 10.252511 -0.0329321231
## V37  821.6538  4.474234  4.275971 -0.0322506678
```

```r
#Making t-test on the covariates

t.test(df1$Time)
```

```
##
##  One Sample t-test
##
## data:  df1$Time
## t = 2.6232, df = 36, p-value = 0.01269
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.003711374 0.029008405
## sample estimates:
##  mean of x
## 0.01635989
```

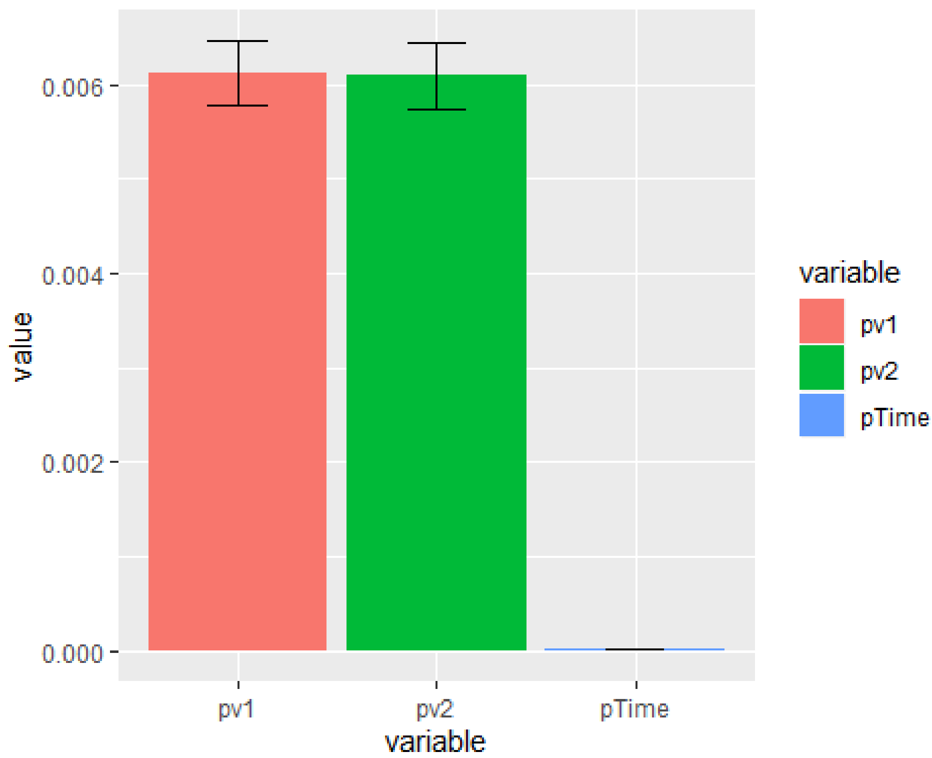
The t-test shows that there is a significant difference between the sum of the slopes for the added covariate time and 0. Thus, time as a covariate is a significant predictor of changes in voxel values across participants, $t(36)=2.62$, $p<0.05$. One possible explanation could be that some of the participants fell asleep during the experiment, which changed the values of the voxels over time.

8. Make a bar diagram like in 6.c., but display effects as percent signal change (hint: percent signal change is slope divided by intercept).

```r
#making new colloumn in df1
df2 <- data.frame("pv1" = df1$V1/df1$Intercept,
                  "pv2" = df1$V2/df1$Intercept,
                  "pTime" = df1$Time/df1$Intercept)




melt_df2 <- melt(df2)
```

```
## No id variables; using all as measure variables
```

```r
ggplot(melt_df2, aes(variable,value, fill = variable))+
  stat_summary(fun.y = mean, geom = "Bar") +
  stat_summary(fun.data = mean_se, geom = "errorbar", width = 0.3)
```


0.006
0.004
0.002
0.000
value
pv1
pv2
pTime
variable
variable
pv1
pv2
pTime